

REMARKS

The examiner is thanked for the performance of a thorough search.

By this amendment, no claims have been amended, cancelled or added. Hence, Claims 1-32 are pending in the application.

OBJECTIONS/REJECTIONS RAISED IN THE OFFICE ACTION

The Abstract was objected to for exceeding 150 words.

Claims 1-32 were rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent No. 6,473,791 issued to Al-Ghosein et al. ("*Al-Ghosein*") in view of U.S. Patent No. 6,230,312 issued to Hunt ("*Hunt*").

The rejections are respectfully traversed.

THE AMENDED ABSTRACT IS 150 WORDS OR LESS

The Abstract is amended herein to recite 150 words or less. Consequently, it is respectfully submitted that the objection to the Abstract has been overcome.

THE PENDING CLAIMS ARE PATENTABLE OVER THE CITED ART

Assuming, *arguendo*, that *Al-Ghosein* and *Hunt* were to be properly combined, each of pending Claims 1-32 would still recite one or more elements that are not disclosed, taught, or suggested by the cited art, either individually or in combination.

Claim 1

Claim 1 recites:

A method for applying one or more policy constraints in an application program, the method comprising the computer-implemented step of redirecting a request to invoke a routine contained in the application program to a policy broker without modifying program code contained in the application program that invokes the routine, wherein the processing of the request to invoke the routine by the policy broker causes the one or more policy constraints to be applied to invocation of the routine. (emphasis added)

At least the features of Claim 1 underlined above are not shown by *Al-Ghosein* or *Hunt*, either individually or in combination.

Claim 1 provides an approach for applying one or more policy constraints in an application program. According to Claim 1, a request to invoke a routine, contained in an application program, is redirected to a policy broker. The request to invoke the routine is redirected to the policy broker without modifying the program code contained in the application program that invokes the routine. The processing of the request to invoke the routine by the policy broker causes one or more policy constraints to be applied to the invocation of the routine. Advantageously, the application programmer does not need to know what policy routines need to be called, and under what conditions, to apply a particular policy constraint in the execution of an application when writing the application program code.

On the other hand, *Al-Ghosein* teaches an approach for providing a centralized security facility (the intelligent trust manager) for implementing security policies. According to *Al-Ghosein*, applications create a request describing an action that needs to be checked against an appropriate security policy. For example, the application may need to determine whether a potentially dangerous action (such as downloading content from a website) that it is proposing to take is allowed or forbidden in accordance with a policy. The application invokes a method of

the intelligent trust manager requesting a decision. An answer is provided to the application, such as “Yes” or “No,” and optionally a rationale for the decision may be provided. After the application obtains the policy decision, the application executes logic based on the obtained policy decision to apply the policy, e.g., if the proposed action is allowed, the proposed action is performed. (See Abstract; Col. 4, line 63- Col. 5, line15; Col. 5, line 60 – Col. 6, line 5.) Thus, in *Al-Ghosein*, the application determines what action should be taken and how a policy is to be applied.

Unfortunately, *Al-Ghosein* suffers from the same deficiencies as prior approaches discussed in the Applicant’s background (e.g., see paragraph 9). Specifically, *Al-Ghosein* requires that the application programmer know what policy routines of the intelligent trust manager need to be called, and under what conditions, to apply a particular policy constraint in the execution of an application when writing the application program code. To illustrate, *Al-Ghosein* states:

To obtain a decision, the application 60 bundles action information includes a name or the like identifying the desired action and policy-specific arguments in to a request 68 (e.g., a COM request object), and invokes a method of the intelligent trust manager 62 requiring a decision. (Col. 5, lines 5-9).

Another problem discussed in the Applicant’s background from which *Al-Ghosein* suffers is that the application programmer must also modify the application program code to include program logic to process the return codes provided by the policy routines. For example, *Al-Ghosein* states:

the decision [the policy decision] is returned to the system component, which then operates accordingly. (See Abstract).

In view of the fundamental differences between the features of Claim 1 and the teachings of *Al-Ghosein*, the element of “wherein the processing of the request to invoke the routine by the

policy broker causes the one or more policy constraints to be applied to invocation of the routine,” featured in Claim 1, is not disclosed, taught, or suggested by *Al-Ghosein*. The portion of *Al-Ghosein* cited to show this feature (Col. 4, lines 17-23) is relied upon to show a policy object making an advisory action (e.g., yes or no), and returning the decision to the system component via the intelligent trust manager. However, this teaching fails to show a policy broker causing one or more policy constraints to be applied to the invocation of the routine when the request to invoke the routine is processed by a policy broker.

Instead, the *Al-Ghosein* teaches obtaining a policy decision in a manner that is completely independent from the invocation of the routine on which the policy constraints are to be applied. For example, *Al-Ghosein* teaches that an application obtains a policy decision from the intelligent trust manager before invoking a routine. After obtaining the policy decision, the application thereafter executes logic according to the obtained policy decision, which may or may not include the invocation of a routine. Thus, when the request to invoke the routine is processed according to *Al-Ghosein*, a policy constraint is not applied to the invocation of the routine because, to the extent that a policy needed to be checked or consulted, the policy was checked or consulted prior to the invocation of the routine.

Consequently, this element is not disclosed, taught, or suggested by *Al-Ghosein*.

Additionally, the Office Action acknowledges that the approach of *Al-Ghosein* does not specifically teach, “without modifying program code,” and *Hunt* is relied upon to show this feature. *Hunt* is directed towards determining how to distribute one or more portions (called units) of an application on one or more computers. A unit of the application may need to reside on a specific computer, and thus, the unit may be subject to a per-unit location constraint. (See Col. 2., lines 26-35).

The portion of *Hunt* cited to show “without modifying program code” (Col. 37, lines 22-29) discusses how to develop a model of distributing units across a group of computers. In this portion, *Hunt* states, “the programmer creates a distributed application with minimal effort simply by running the application through profiling scenarios and writing the corresponding distribution model into the application binary without modifying application sources.” Importantly, the phrase “without modifying application sources,” in this context, does not refer to not modifying program code (software), but instead refers to not changing the computer to which a unit of the application is assigned in a distribution model. Indeed, the only mention of program code in the cited portion of *Hunt* is the suggestion that the application binary may be modified by writing a distribution model into the application binary. Thus, to the extent that *Hunt* discloses program code, *Hunt* actually discloses modifying the program code. It is further submitted that *Hunt* contains no suggestion of redirecting a request to invoke a routine contained in an application program to a policy broker.

As a result, even if *Al-Ghosein* and *Hunt* were to be combined, the resulting combination would still fail to suggest “redirecting a request to invoke a routine contained in the application program to a policy broker without modifying program code contained in the application program that invokes the routine” as featured Claim 1. As a result, this element is not disclosed, taught, or suggested by *Al-Ghosein* and *Hunt*, either individually or in combination.

Moreover, *Al-Ghosein* and *Hunt* have not been properly combined. The Office Action states that it would have been obvious:

to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Mohsen Al-Ghosein with *Hunt* because *Hunt*’s teachings would allowed [sic] a programmer to insert or remove constraints on a specific application without changing the application sources. Therefore, reducing the cost and improving performance and efficiency of Mohsen Al-Ghosein’s system.

However, notwithstanding the fact that *Al-Ghosein* and *Hunt* do not disclose numerous claim features, the Applicant respectfully submits that there is nothing in either *Al-Ghosein* or *Hunt* that teaches or suggests combining their respective teachings.

As stated in the Federal Circuit decision *In re Dembiczak*, 50 USPQ.2d 1617 (Fed. Cir. 1999), (citing *Gore v. Garlock*, 220 USPQ 303, 313 (Fed. Cir. 1983)), “it is very easy to fall victim to the insidious effect of the hindsight syndrome where that which only the inventor taught is used against its teacher.” *Id.* The Federal Circuit stated in *Dembiczak* “that the best defense against subtle but powerful attraction of a hindsight-based obviousness analysis is rigorous application of the requirement for a showing of the teaching or suggestion to combine prior art references.” *Id.* Thus, the Federal Circuit explains that a proper obviousness analysis requires “**particular factual findings** regarding the locus of the suggestion, teaching, or motivation to combine prior art references.” *Id.* (emphasis added).

In particular, the Federal Circuit states:

“We have noted that evidence of a suggestion, teaching, or motivation to combine may flow from the prior art references themselves, the knowledge of one of ordinary skill in the art, or, in some cases, from the nature of the problem to be solved...although ‘the suggestion more often comes from the teachings of the pertinent references’...The range of sources available, however, does **not diminish the requirement for actual evidence**. That is, the **showing must be clear and particular**...Broad conclusory statements regarding the teaching of multiple references, standing alone, are not ‘evidence.’” *Id.* (emphasis added; internal citations omitted).

Neither *Al-Ghosein* nor *Hunt* show any suggestion, teaching, or motivation to combine their teachings, nor does the Office Action provide a “clear and particular” showing of the suggestion, teaching, or motivation to combine their teachings. The Office Action does not cite a portion of *Al-Ghosein* or *Hunt* to show a motivation to combine their teachings. In fact, the only motivation provided in the Office Action is the hindsight observation that by combining features

of those references, one may achieve the benefits achieved from the invention as described and claimed in the application.

Further, in view of the above-discussed mischaracterization of *Hunt*'s teachings, it is entirely unclear how *Al-Ghosein* may be combined for use with *Hunt*. For example, no portion of *Al-Ghosein* suggests how the applications in the approach of *Al-Ghosein* may be decomposed into units for placement on two or computers in a distributed system. Similarly, no portion of *Hunt* suggests how a centralized security facility of *Al-Ghosein* could be implemented in the distributed model of *Hunt*. In fact, the two references appear to teach away from each other, as both references discuss two entirely different paradigms (*Al-Ghosein* is directed towards providing a centralizing source of policy information, whereas *Hunt* is directed towards providing a decentralized or distributed application).

It is respectfully submitted that such a hindsight observation is not consistent with the Federal Circuit's requirement for "particular factual findings."

For at least the above reasons, it is respectfully submitted that one or more elements of Claim 1 are not disclosed, taught, or suggested by *Al-Ghosein* or *Hunt*, considered either individually or in combination. Further, it is respectfully submitted that *Al-Ghosein* and *Hunt* have not been properly combined, and thus, a rejection of Claim 1 under 35 U.S.C. § 103(a) cannot be supported. Consequently, Claim 1 is patentable over the cited art, and is in condition for allowance.

Claim 15

Claim 15 recites:

A method for implementing policy constraints in an application program, the method comprising the computer-implemented steps of:

identifying a routine in the application program for which one or more policy constraints are to be applied, wherein the routine is invoked by program code contained in the application program; and
without modifying the program code, substituting replacement code for original code contained in the identified routine, wherein execution of the replacement code by one or more processors causes the one or more policy constraints to be applied.(emphasis added)

At least the features of Claim 15 underlined above are not shown by *Al-Ghosein* or *Hunt*, either individually or in combination.

No portion of *Al-Ghosein* teaches, “without modifying the program code, substituting replacement code for original code contained in the identified routine, wherein execution of the replacement code by one or more processors causes the one or more policy constraints to be applied.” Instead, the cited portion (Col. 4, lines38-46) merely states, *in toto*:

To replace a policy with another policy, the other policy is registered and the name binding (described below) changed so those system components using the name invoke the other policy instead of the existing policy. Among other benefits, the present invention thus allows policies to be shared by numerous other system components from the details of policy administration.

This portion lacks any suggestion of substituting replacement code for original code contained in the identified routine. At best, this portion suggests that (a) the policy which a system component, such as an application, checks before invoking a routine may be changed, and (b) a policy may be used (or shared) by two or more system components. However, this is not analogous to substituting replacement code for original code contained in the identified routine. Instead, *Al-Ghosein* teaches that the original code contained in the routine is in no way modified when a policy constraint is applied. In fact, according to *Al-Ghosein*, a policy is checked by making a call to an intelligent trust manager prior to the invocation of the routine. Consequently, this element is not disclosed, taught, or suggested by *Al-Ghosein*.

For at least the above reasons, it is respectfully submitted that one or more elements of

Claim 15 is not disclosed, taught, or suggested by *Al-Ghosein* or *Hunt*, either individually or in combination. Further, as explained above, it is respectfully submitted that *Al-Ghosein* and *Hunt* have not been properly combined, and thus, a rejection of Claim 15 under 35 U.S.C. § 103(a) cannot be supported. Consequently, Claim 15 is patentable over the cited art, and is in condition for allowance.

Claim 27

Claim 27 recites:

An apparatus for implementing one or more policy constraints in an application program, the apparatus comprising:
a memory; and
a code substitution mechanism communicatively coupled to the memory and being configured to, for a routine in an application program invoked by program code in the application program, substituting, without modifying the program code, original code contained in the identified routine with replacement code, wherein execution of the replacement code by one or more processors causes the one or more policy constraints to be applied.(emphasis added)

At least the features of Claim 27 underlined above are not shown by *Al-Ghosein* or *Hunt*, either individually or in combination.

Claim 27 has been rejected under the same rationale as Claim 15. However, it is submitted that Claim 15 is patentable over the cited art since numerous features of Claim 15 are not disclosed, taught, or suggested by *Al-Ghosein* or *Hunt*, either individually or in combination. Thus, for at least the reasons given above, it is respectfully submitted that no portion of *Al-Ghosein* or *Hunt*, either individually or in combination, discloses, teaches, or suggests the element of “substituting, without modifying the program code, original code contained in the identified routine with replacement code, wherein execution of the replacement code by one or more processors causes the one or more policy constraints to be applied” recited in Claim 27.

It is respectfully submitted that one or more elements of Claim 27 is not disclosed, taught, or suggested by *Al-Ghosein* or *Hunt*, either individually or in combination. Further, as explained above, it is respectfully submitted that *Al-Ghosein* and *Hunt* have not been properly combined, and thus, a rejection of Claim 27 under 35 U.S.C. § 103(a) cannot be supported. Consequently, Claim 27 is patentable over the cited art, and is in condition for allowance.

Claims 2-14, 16-26, and 28-32

Claims 8 and 21 each recite features similar to those discussed above with respect to Claims 1 and 15 respectively, except that they are recited in computer-readable medium format. Consequently, it is respectfully submitted that Claims 8 and 21 are patentable over the cited art and are each in condition for allowance for at least the reasons given above with respect to Claims 1 and 15.

Claims 2-7, 9-14, 16-20, 22-26, and 28-32 are dependent claims, each of which depends (directly or indirectly) on one of the claims discussed above. Each of Claims 2-7, 9-14, 16-20, 22-26, and 28-32 is therefore allowable for the reasons given above for the claim on which it depends. In addition, each of Claims 2-7, 9-14, 16-20, 22-26, and 28-32 introduce one or more additional limitations that independently render it patentable. However, due to the fundamental differences already identified, to expedite the positive resolution of this case a separate discussion of those limitations is not included at this time, although the Applicant reserves the right to further point out the differences between the cited art and the novel features recited in the dependent claims.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

A petition for extension of time, to the extent necessary to make this reply timely filed, is hereby made. If any applicable fee is missing or insufficient, throughout the pendency of this application, the Commissioner is hereby authorized to charge any applicable fees and to credit any overpayments to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP



Christopher J. Brokaw
Reg. No. 45,620
Date: May 23, 2005

2055 Gateway Place, Suite 550
San Jose, CA 95110
(408) 414-1225
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: **Mail Stop Amendment**, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

on May 23, 2005

by


Angelica Maloney